

---

# **NEC ID Documentation**

*Release 1.1*

**NEC Australia**

**Aug 07, 2019**



<b>1</b>	<b>Using the API</b>	<b>3</b>
1.1	Authentication and Authorisation . . . . .	3
1.2	Biometric REST Endpoints . . . . .	4
1.3	Tenant Management REST Endpoints . . . . .	28
1.4	Error Handling . . . . .	39
<b>2</b>	<b>NEC ID SDKs</b>	<b>41</b>
2.1	Platforms . . . . .	41
	<b>HTTP Routing Table</b>	<b>43</b>



NEC ID provides biometrics as a service.



NEC ID exposes a secure REST (Representational State Transfer) API, allowing users to interact with the NEC ID biometric service over HTTPS. JSON is returned by all API responses including errors and HTTP response status codes are used to designate success and failure.

The platform also provides a means to manage your tenancy configuration via secure REST API, allowing tenants to provision new galleries and applications.

The following documentation details the authentication and authorisation requirements and the various endpoints.

Download NEC ID API Swagger [JSON](#) or [YAML](#)

## 1.1 Authentication and Authorisation

NEC ID authentication requires and AWS Signature Version 4 authorization header to be included with each request. This requires the **AccessKey** and **SecretKey** provided in the NEC ID Welcome Pack and the use of the `ap-southeast-2` **AWS Region** and `execute-api` **Service name**.

The provided *NEC ID SDKs* handle the signing process using your parameters and tools such as Postman have an in-built AWS Signature authorisation type.

Alternatively this header can be generated using the [Amazon's authenticating requests guide](#).

The following is an example of a request with the **Authorization** and **x-amz-date** header values. Line breaks are added to this Authorization line for readability.

In addition, an application API Key must be included in requests that pertain to a specific gallery by using the **x-api-key** header to identify the target application.

### Example request:

```
POST /v1.1/subjects HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
x-api-key: 73n4n74p1k3y
x-amz-date: 20180816T032046Z
Authorization: AWS4-HMAC-SHA256
Credential=AKIAIOSFODNN8EXAMPLE/20190304/ap-southeast-2/execute-api/aws4_
↪request,
SignedHeaders=content-length;content-type;host;x-amz-date;x-api-key,
Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024
```

Failure to include a valid Authorization header will result in a 403 forbidden HTTP response status code, with a JSON message describing the issue with the signature.

**Example response:**

```
HTTP/1.1 403 Forbidden
Content-Type: application/json

{
  "message": "Signature expired: 20180816T032046Z is now earlier than_
↪20180816T040329Z (20180816T040829Z - 5 min.)"
}
```

Failure to include a valid API Key via the x-api-key header in particular requests will result in a 403 forbidden HTTP response status codes, with a JSON message of Forbidden.

**Example response:**

```
HTTP/1.1 403 Forbidden
Content-Type: application/json

{
  "message": "Forbidden"
}
```

## 1.2 Biometric REST Endpoints

NEC ID support endpoints for:

- registering, updating, listing and unregistering subjects;
- registering, updating, listing and unregistering subject events;
- performing face extraction, search and verification;
- creating, renaming, deleting and retrieving tags;
- performing a bulk subject registration;

### 1.2.1 Subjects

Manage subjects for search and verification matching.

#### List Subjects

**GET /v1.1/subjects**

Retrieve all subjects for a specific gallery.



**Example request:**

```
GET /v1.1/subjects HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

**Query Parameters**

- **start** – Optional starting index of the request, defaults to 0 if not provided.
- **length** – Optional length of the list, defaults to 1000 if not provided. Limited to a maximum of 1000 per request.
- **dir** – Optional last updated sort direction (asc or desc). default is asc.

**Request Headers**

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "subjects": [
    {
      "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b",
      "eventId": "eventguid-caf3-4e0f-92b9-101a9e73a3ee",
      "status": "ACTIVE",
      "lastUpdated": "2018-12-18T00:27:56.427+00:00"
    },
    {
      "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b",
      "eventId": "eventguid-f99a-41dc-8eb1-cd7b1b3dcdec",
      "status": "DELETED",
      "lastUpdated": "2019-02-28T18:11:49.157+00:00"
    },
    {
      "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b",
      "eventId": "eventguid-f99a-41dc-8eb1-cd7b1b3dcdec",
      "status": "ACTIVE",
      "lastUpdated": "2019-02-28T18:11:49.174+00:00"
    }
  ],
  "total": 3
}
```

### Response JSON Object

- **subjects** (*array*) – Containing **id** (*string*): Subject id, **eventId** (*string*): Event id, **status** (*string*): Status, **lastUpdated** (*ISO 8601 timestamp*): Last Updated Time.
- **total** (*int*) – Total number of events in gallery.

### Response Headers

- **Content-Type** – application/json

### Status Codes

- **200 OK** – OK.
- **404 Not Found** – Subjects not found.

---

**Note:** As a subject can have many events, the subject's id will be repeated for each of its eventIds. Plus, in order to reuse slots within a gallery, deleted events are flagged with a **status** of DELETED until the slot is recycled. This means that the list of subjects can include deleted subjects, plus subjects we have been updated, which appear as deleted and active with the same ids.

---

## Register Subject

### POST /v1.1/subjects

Registers a new subject.

#### Example request:

```
POST /v1.1/subjects HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "face": "[FACE]",
  "tags": [ "passport" ]
}
```

### Request JSON Object

- **face** (*string*) – Base64 encoded image.
- **tags** (*array*) – Optional list of tag names to register against the subject.

### Request Headers

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.

- Authorization – AWS Signature Version 4.

**Example response:**

```

HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b",
  "eventId": "eventguid-caf3-4e0f-92b9-101a9e73a3ee",
  "attributes": {
    "faceArea": {
      "left": "161",
      "top": "156",
      "right": "318",
      "bottom": "313"
    },
    "headArea": {
      "left": "131",
      "top": "86",
      "right": "345",
      "bottom": "343"
    },
    "rightEyeCenter": {
      "x": "195",
      "y": "198"
    },
    "leftEyeCenter": {
      "x": "276",
      "y": "191"
    },
    "frontalFaceScore": "0.569824",
    "faceRoll": "4.93922",
    "facePan": "-9.34863",
    "faceTilt": "7.613",
    "faceScore": "0.999591",
    "faceQualityScore": "0.86541"
  }
}

```

**Response JSON Object**

- **id** (*string*) – Subject id.
- **eventId** (*string*) – Event id.
- **attributes** – See *Face Attributes*.

**Response Headers**

- Content-Type – application/json

**Status Codes**

- 201 Created – Subject has been created.

## Update Subject

**PUT** `/v1.1/subjects/` (**string**: `subjectId`)  
 Update an existing subject.

### Example request:

```
PUT /v1.1/subjects/necidguid-fcdf-49eb-9182-5a6825ed2a3b HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "face": "[FACE]",
  "tags": [ "passport" ]
}
```

### Parameters

- **subjectId** – Subject id.

### Request JSON Object

- **face** (*string*) – Base64 encoded image.
- **tags** (*array*) – Optional list of tag names to register against the subject.

### Request Headers

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b",
  "eventId": "eventguid-caf3-4e0f-92b9-101a9e73a3ee",
  "attributes": {
    "faceArea": {
      "left": "161",
      "top": "156",
      "right": "318",
      "bottom": "313"
    },
    "headArea": {
      "left": "131",
```

(continues on next page)

(continued from previous page)

```

        "top": "86",
        "right": "345",
        "bottom": "343"
    },
    "rightEyeCenter": {
        "x": "195",
        "y": "198"
    },
    "leftEyeCenter": {
        "x": "276",
        "y": "191"
    },
    "frontalFaceScore": "0.569824",
    "faceRoll": "4.93922",
    "facePan": "-9.34863",
    "faceTilt": "7.613",
    "faceScore": "0.999591",
    "faceQualityScore": "0.86541"
}
}

```

**Response JSON Object**

- **id** (*string*) – Subject id.
- **eventId** (*string*) – Event id.
- **attributes** – See *Face Attributes*.

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- **200 OK** – Subject has been updated.
- **404 Not Found** – Subject with id not found.

**Unregister Subject****DELETE** /v1.1/subjects/ (**string:** *subjectId*)

Unregister an existing subject and related events.

**Example request:**

```

DELETE /v1.1/subjects/necidguid-fcdf-49eb-9182-5a6825ed2a3b HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

```

**Parameters**

- **subjectId** – Subject id.

**Request Headers**

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```
HTTP/1.1 204 No Content
Content-Type: application/json
```

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- **204 No Content** – Subject has been unregistered.
- **404 Not Found** – Subject with id not found.

## 1.2.2 Events

Events provides the ability to register multiple biometrics events for a subject. The API allows you to, list register, update and unregister subject events.

### List Events

**GET /v1.1/subjects/ (string: *subjectId*) /events**

Retrieve all events for a given subject.

**Example request:**

```
GET /v1.1/subjects/necidguid-fcdf-49eb-9182-5a6825ed2a3b/events HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

**Parameters**

- **subjectId** – Subject id.

**Request Headers**

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.

- *x-amz-date* – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- *Authorization* – AWS Signature Version 4.

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b",
  "events": [
    {
      "id": "eventguid-caf3-4e0f-92b9-101a9e73a3ee"
    },
    {
      "id": "eventguid-f99a-41dc-8eb1-cd7b1b3dcdec"
    }
  ]
}
```

**Response JSON Object**

- **id** (*string*) – Subject id.
- **events** (*array*) – Containing **id** (*string*): Event id.

**Response Headers**

- *Content-Type* – application/json

**Status Codes**

- 200 OK – OK.
- 404 Not Found – Subject with id not found.

**Register Event****POST /v1.1/subjects/(string: *subjectId*)/events**

Register a new event for a given subject.

**Example request:**

```
POST /v1.1/subjects/necidguid-fcdf-49eb-9182-5a6825ed2a3b/events HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "face": "[FACE]",
  "tags": [ "licence" ]
}
```

**Request JSON Object**

- **face** (*string*) – Base64 encoded image.

- **tags** (*array*) – Optional list of tag names to register against the subject.

**Parameters**

- **subjectId** – Subject id.

**Request Headers**

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```

HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b",
  "eventId": "eventguid-f99a-41dc-8eb1-cd7b1b3dcdec",
  "attributes": {
    "faceArea": {
      "left": "161",
      "top": "156",
      "right": "318",
      "bottom": "313"
    },
    "headArea": {
      "left": "131",
      "top": "86",
      "right": "345",
      "bottom": "343"
    },
    "rightEyeCenter": {
      "x": "195",
      "y": "198"
    },
    "leftEyeCenter": {
      "x": "276",
      "y": "191"
    },
    "frontalFaceScore": "0.569824",
    "faceRoll": "4.93922",
    "facePan": "-9.34863",
    "faceTilt": "7.613",
    "faceScore": "0.999591",
    "faceQualityScore": "0.86541"
  }
}

```

**Response JSON Object**

- **id** (*string*) – Subject id.



- **eventId** (*string*) – Event id.
- **attributes** – See *Face Attributes*.

### Response Headers

- **Content-Type** – application/json

### Status Codes

- **201 Created** – Event has been created.
- **404 Not Found** – Subject with id not found.

## Update Event

**PUT** `/v1.1/subjects/{string: subjectId}/events/`  
**string:** *eventId* Update an existing event for a given subject.

### Example request:

```
PUT /v1.1/subjects/necidguid-fcdf-49eb-9182-5a6825ed2a3b/events/eventguid-f99a-
↪41dc-8eb1-cd7b1b3dcdec HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "face": "[FACE]",
  "tags": [ "licence" ]
}
```

### Request JSON Object

- **face** (*string*) – Base64 encoded image.
- **tags** (*array*) – Optional list of tag names to register against the subject.

### Parameters

- **subjectId** – Subject id.
- **eventId** – Event id.

### Request Headers

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

### Example response:

```

HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b",
  "eventId": "eventguid-f99a-41dc-8eb1-cd7b1b3dcdec",
  "attributes": {
    "faceArea": {
      "left": "161",
      "top": "156",
      "right": "318",
      "bottom": "313"
    },
    "headArea": {
      "left": "131",
      "top": "86",
      "right": "345",
      "bottom": "343"
    },
    "rightEyeCenter": {
      "x": "195",
      "y": "198"
    },
    "leftEyeCenter": {
      "x": "276",
      "y": "191"
    },
    "frontalFaceScore": "0.569824",
    "faceRoll": "4.93922",
    "facePan": "-9.34863",
    "faceTilt": "7.613",
    "faceScore": "0.999591",
    "faceQualityScore": "0.86541"
  }
}

```

### Response JSON Object

- **id** (*string*) – Subject id.
- **eventId** (*string*) – Event id.
- **attributes** – See *Face Attributes*.

### Response Headers

- **Content-Type** – application/json

### Status Codes

- **200 OK** – Event has been updated.
- **404 Not Found** – Subject with id and eventId not found.

## Unregister Event

**DELETE** /v1.1/subjects/(string: *subjectId*)/events/  
 string: *eventId* Unregister an existing event for a given subject.

Example request:

```
DELETE /v1.1/subjects/necidguid-fcdf-49eb-9182-5a6825ed2a3b/events/eventguid-f99a-
↪41dc-8eb1-cd7b1b3dcdec HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

**Parameters**

- **subjectId** – Subject id.
- **eventId** – Event id.

**Request Headers**

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```
HTTP/1.1 204 No Content
Content-Type: application/json
```

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- **204 No Content** – Event has been unregistered.
- **404 Not Found** – Subject with id and eventId not found.

## 1.2.3 Face

Extract faces, search and verify probe images against registered subjects.

**Extract****POST /v1.1/face/extract**

Extract face attributes.

**Example request:**

```
POST /v1.1/face/extract HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "faces": "[FACES]"
}
```

### Request JSON Object

- **faces** (*string*) – Base64 encoded image containing one or more faces.

### Request Headers

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "faces": [
    {
      "attributes": {
        "faceArea": {
          "left": "434",
          "top": "449",
          "right": "503",
          "bottom": "518"
        },
        "headArea": {
          "left": "423",
          "top": "418",
          "right": "515",
          "bottom": "531"
        },
        "rightEyeCenter": {
          "x": "451",
          "y": "464"
        },
        "leftEyeCenter": {
          "x": "487",
          "y": "466"
        },
        "frontalFaceScore": "0.574219",
        "faceRoll": "-3.17983",
        "facePan": "-9.72147",
      }
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

        "faceTilt": "7.93723",
        "faceScore": "0.999993",
        "faceQualityScore": "0.749248"
    },
    {
        "attributes": {
            "faceArea": {
                "left": "850",
                "top": "416",
                "right": "921",
                "bottom": "486"
            },
            "headArea": {
                "left": "838",
                "top": "383",
                "right": "933",
                "bottom": "500"
            },
            "rightEyeCenter": {
                "x": "866",
                "y": "431"
            },
            "leftEyeCenter": {
                "x": "904",
                "y": "431"
            },
            "frontalFaceScore": "0.574707",
            "faceRoll": "-0",
            "facePan": "8.36667",
            "faceTilt": "1.05248",
            "faceScore": "0.999994",
            "faceQualityScore": "0.764253"
        }
    }
]
}

```

**Response JSON Object**

- **faces** (array) – Containing **attributes**: See *Face Attributes*.

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- **200 OK** – Face attributes extracted.

**Search**

**POST /v1.1/face/search**

Search for subjects using a probe.

**Example request:**

```

POST /v1.1/face/search HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "probe": "[PROBE]",
  "threshold": 7500,
  "limit": 3,
  "tags": [ "passport", "licence" ]
}

```

### Request JSON Object

- **probe** (*string*) – Base64 encoded image.
- **threshold** (*int*) – Optional score threshold, ranges from 0 to 9999, default is 7000.
- **limit** (*int*) – Optional limit of total events returned, ranges from 1 to 50, default is 10.
- **tags** (*array*) – Option list of tag names to refine the search against, using OR to filter subjects.

### Request Headers

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

### Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "candidates": [
    {
      "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b",
      "score": 8200,
      "events": [
        {
          "id": "eventguid-caf3-4e0f-92b9-101a9e73a3ee"
          "score": 8200
        },
        {
          "id": "eventguid-f99a-41dc-8eb1-cd7b1b3dcdec"
          "score": 8000
        }
      ]
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

    },
    {
      "id": "necidguid-0d05-4052-a44f-83f6b243e70b",
      "score": 7600,
      "events": [
        {
          "id": "eventguid-f8bc-47d1-a976-7e8b953da664"
          "score": 7600
        }
      ]
    }
  ],
  "attributes": {
    "faceArea": {
      "left": "161",
      "top": "156",
      "right": "318",
      "bottom": "313"
    },
    "headArea": {
      "left": "131",
      "top": "86",
      "right": "345",
      "bottom": "343"
    },
    "rightEyeCenter": {
      "x": "195",
      "y": "198"
    },
    "leftEyeCenter": {
      "x": "276",
      "y": "191"
    },
    "frontalFaceScore": "0.569824",
    "faceRoll": "4.93922",
    "facePan": "-9.34863",
    "faceTilt": "7.613",
    "faceScore": "0.999591",
    "faceQualityScore": "0.86541"
  }
}

```

**Response JSON Object**

- **candidates** (*array*) – Containing **id** (*string*): Subject id, **score** (*int*): Highest match score and **event** (*array*): Containing **id** (*string*): Event id and **score** (*int*): Match score.
- **attributes** – See *Face Attributes*.

**Response Headers**

- Content-Type – application/json

**Status Codes**

- 200 OK – Candidates found.
- 404 Not Found – Candidates not found.

## Verify

### POST /v1.1/face/verify

Verify a subject, and their one or more events, against a probe.

#### Example request:

```
POST /v1.1/face/verify HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "probe": "[PROBE]",
  "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b"
}
```

#### Request JSON Object

- **probe** (*string*) – Base64 encoded image.
- **id** (*string*) – Subject id.

#### Request Headers

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

#### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "necidguid-fcdf-49eb-9182-5a6825ed2a3b",
  "score": 8200
}
```

#### Response JSON Object

- **id** (*string*) – Subject id.
- **score** (*int*) – Match score of the subject's highest scoring event.

#### Response Headers

- **Content-Type** – application/json

#### Status Codes

- **200 OK** – Subject verified.



- 404 Not Found – Subject not verified.

## 1.2.4 Tags

Create, update, delete and retrieve tags. Tags provide the ability to tag subjects and events. You can then search for subjects based on these tags.

Tags must be created first, before using them in register and search requests.

### List Tags

#### GET /v1.1/tags

Retrieve all tags for a specific gallery.

#### Example request:

```
GET /v1.1/tags HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

#### Request Headers

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

#### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  "passport",
  "licence"
]
```

#### Response JSON Object

- **array** – Containing tag names.

#### Response Headers

- **Content-Type** – application/json

#### Status Codes

- 200 OK – OK.

---

**Note:** Each gallery has a hard limit of 64 tags

---

### Create Tag

**POST** `/v1.1/tags/` (**string:** *name*)

Create a tag.

**Example request:**

```
POST /v1.1/tags/staff HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

#### Parameters

- **name** – The tag name. Tag name must not be empty and must be unique.

#### Request Headers

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json
```

#### Response Headers

- **Content-Type** – application/json

#### Status Codes

- **200 OK** – Tag created.

### Update Tag

**PUT** `/v1.1/tags/` (**string:** *oldName*)

Update a tag.

**Example request:**

```
POST /v1.1/tags/licence HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "newName": "identification"
}
```

**Request JSON Object**

- **newName** (*string*) – The tag’s new name. Tag name must not be empty and must be unique.

**Parameters**

- **oldName** – The tag’s current name.

**Request Headers**

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json
```

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- **200 OK** – Tag updated.

**Delete Tag**

**DELETE** /v1.1/tags/ (**string:** *name*)

Deletes a tag. Note that existing subjects and events are not updated.

**Example request:**

```
DELETE /v1.1/tags/passport HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
```

(continues on next page)

(continued from previous page)

```
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

### Parameters

- **name** – The name of the tag to delete.

### Request Headers

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

### Response Headers

- **Content-Type** – application/json

### Status Codes

- **200 OK** – Tag deleted.

## 1.2.5 Jobs

Jobs allow you to perform bulk operations on your gallery such as bulk registrations.

### Bulk Register

#### POST /v1.1/jobs/bulkregister

Creates a job to run bulk registration against the supplied registrations.

#### Example request:

```
POST /v1.1/jobs/bulkregister HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "registrations": [
    {
      "filename": "1.jpg",
```

(continues on next page)

(continued from previous page)

```

    "tags": [ "passport" ]
  },
  {
    "filename": "2.jpg",
    "tags": [ "passport" ]
  }
]
}

```

**Request JSON Object**

- **registrations** (*array*) – Containing **filename** (*string*): Name of file in S3 bucket and **tags** (*array*): List of tags.

**Request Headers**

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "batchId": "batchguid-eec5-440a-89fc-60817f5546c8"
}

```

**Response JSON Object**

- **batchId** (*string*) – Batch id.

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- **200 OK** – Job created.

**Bulk Register Progress**

**GET** /v1.1/jobs/bulkregister/ (**string:** *batchId*) /  
**string:** *pagingId* Retrieves the progress of the batch operation. In the event of the batch operation having more records than the maximum page size (1000), the results will be paged and the lastEvaluatedKey will contain a value that needs to be sent as the pagingId to retrieve the next page of data.

**Example request:**

```
GET /v1.1/jobs/bulkregister/batchguid-eec5-440a-89fc-60817f5546c8 HTTP/1.1
Host: api.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-api-key: [Application API key]
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

### Parameters

- **batchId** – The batchId of the job.
- **pagingId** – (Optional) The lastEvaluatedKey from the previous request.

### Request Headers

- **Host** – api.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-api-key** – Application API Key.
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "batchId": "batchguid-eec5-440a-89fc-60817f5546c8",
  "processed": [
    {
      "id": "51448BB9-956D-44FD-89AC-A5065D30D084",
      "filename": "1.jpg",
      "attempts": "1",
      "eventId": "eventguid-B190-466E-A1C2-9E85F436775A",
      "subjectId": "necidguid-45EC-4A97-88F0-F064A829FC90",
      "registerStatus": "Success",
      "reason": "",
      "status": "Processed",
      "attributesJson": "..."
    },
    {
      "id": "EFE569DA-AE74-43CC-A75F-BD85B07A7401",
      "filename": "2.jpg",
      "attempts": "1",
      "eventId": "eventguid-F3BE-4428-AAC3-8C9631776364",
      "subjectId": "necidguid-113B-4726-A267-4267816EC8FB",
      "registerStatus": "Failed",
      "reason": "Image too small",
      "status": "Processed",
      "attributesJson": "..."
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```
"lastEvaluatedKey": ""
}
```

**Response JSON Object**

- **batchId** (*string*) – Batch Id.
- **processed** (*array*) – List of records within the batch with a summary. See *Register Request Summaries*
- **lastEvaluatedKey** (*string*) – Key to be sent as the pagingId to retrieve the next page of data.

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- **200 OK** – OK

**1.2.6 Register Request Summaries**

Object describing a register request summary which contains the subjectId and eventId on success.

- id** The operation id. Note this is not the subject or event id.
- filename** The filename from the original request.
- attempts** The number of attempts.
- eventId** The event id on success.
- subjectId** The subject id on success.
- registerStatus** Once the status is “Processed”, this will have a value of “Success” or “Failed”.
- reason** The failure reason on failure.
- status** The status of the operation. Either “New” or “Processed”.
- attributesJson** The Face Attributes - see *Face Attributes*.

**1.2.7 Face Attributes**

Successful requests to the **register**, **update** and **search** endpoints will return an **attributes** property containing a collection of face attributes related to the submitted **face** or **probe** image. In the case of **extract**, the returned **faces** property contains an array of objects with the **attributes** property, representing each face found (up to 20 faces) in the submitted **faces** image.

- faceArea** Pixel coordinates (left, top, right and bottom) defining the bounds of the face.
- headArea** Pixel coordinates (left, top, right and bottom) defining the bounds of the face.
- rightEyeCenter** Pixel coordinates (x and y) of the centre of the right eye.
- leftEyeCenter** Pixel coordinates (x and y) of the centre of the left eye.
- faceScore** Measure of likeness to a face.
- frontalFaceScore** Degree of frontal view of face.

**faceQualityScore** Estimated overall quality of face.

**faceRoll** Roll angle of face in degrees.

**facePan** Pan angle of face in degrees.

**faceTilt** Tilt angle of face in degrees.

---

**Note:** The overall faceQualityScore should be used to determine the quality of the face prior to registration.

---

## 1.3 Tenant Management REST Endpoints

NEC ID support endpoints for:

- creating and deleting galleries and applications with which to access them;

### 1.3.1 Galleries

Manage tenant galleries.

#### List Galleries

**GET** /api/galleries

Retrieve all galleries.

**Example request:**

```
GET /api/galleries HTTP/1.1
Host: portal.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

#### Request Headers

- **Host** – portal.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "galleries": [
    {
```

(continues on next page)



(continued from previous page)

```

    "id": "galleryguid-327D-4D31-8E25-35891A034220",
    "name": "Staff",
    "description": "Staff",
    "size": 10000,
    "count": 3891
  },
  {
    "id": "galleryguid-5570-4f20-a95c-8e50e0bca0cb",
    "name": "Y9Students",
    "description": "Year 9 students",
    "size": 1000,
    "count": 873
  },
  {
    "id": "galleryguid-b751-4b5e-9e4d-cc1a314788db",
    "name": "Visitors",
    "description": "Visitors",
    "size": 10000,
    "count": 431
  }
]
}

```

**Response JSON Object**

- **galleries** (*array*) – Containing **id** (*string*): Gallery id, **name** (*string*): Name, **description** (*string*): Description, **size** (*int*): Size, **count** (*int*): Count.

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- 200 OK – OK.

**Get Gallery**

**GET** /api/galleries/ (**string**: *galleryId*)

Retrieve gallery details, including associated applications and their apiKey.

**Example request:**

```

GET /api/galleries/galleryguid-327D-4D31-8E25-35891A034220 HTTP/1.1
Host: portal.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

```

**Parameters**

- **galleryId** – Gallery id.

**Request Headers**

- **Host** – portal.id.nec.com.au
- **Accept** – application/json

- **Content-Type** – application/json
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "galleryguid-327D-4D31-8E25-35891A034220",
  "name": "Staff",
  "description": "Staff",
  "size": 10000,
  "count": 3891,
  "applications": [
    {
      "id": "applicationId02",
      "name": "Building Access System",
      "apiKey": [Application API key]
    }
  ]
}
```

### Response JSON Object

- **id** (*string*) – Gallery id.
- **name** (*string*) – Name.
- **description** (*string*) – Description.
- **size** (*int*) – Size.
- **count** (*int*) – Count.
- **applications** (*array*) – Containing **id** (*string*): Application id, **name** (*string*): Name, **apiKey** (*string*): API Key.

### Response Headers

- **Content-Type** – application/json

### Status Codes

- 200 OK – OK.

## Create Gallery

### POST /api/galleries

Create a gallery and an application with which to access it.

### Example request:

```
POST /api/galleries HTTP/1.1
Host: portal.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

(continues on next page)

(continued from previous page)

```
{
  "name": "Y10Students",
  "description": "Year 10 Students",
  "faceMinimumQualityScore": 0.8,
  "size": 1000,
}
```

**Request JSON Object**

- **name** (*string*) – Name of the gallery.
- **description** (*string*) – Description of the gallery.
- **faceMinimumQualityScore** (*float*) – Minimum estimated overall quality of face - see *Face Attributes*, must be greater than or equal to 0.45.
- **size** (*int*) – Size of the gallery.

**Request Headers**

- **Host** – portal.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "galleryguid-28da-4dd4-b10b-e48c6be09689",
  "name": "Y10Students",
  "description": "Year 10 Students",
  "size": 1000,
  "apiKey": [Application API key]
}
```

**Response JSON Object**

- **id** (*string*) – Gallery id.
- **name** (*string*) – Name.
- **description** (*string*) – Description.
- **size** (*int*) – Size.
- **apiKey** (*string*) – Application API Key.

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- **200 OK** – Gallery created.

**Note:** If the gallery already exists, a 400 response will be returned, with the error type `Duplicate` and the existing `apiKey` in the message.

---

### Update Gallery

**PATCH** `/api/galleries/` (**string:** `galleryId`)

Update a gallery.

#### Example request:

```
PATCH /api/galleries HTTP/1.1
Host: portal.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "name": "Y10Students",
  "description": "Year 10 Students",
  "size": 2000,
}
```

#### Parameters

- **galleryId** – Gallery id.

#### Request JSON Object

- **name** (*string*) – Name of the gallery.
- **description** (*string*) – Description of the gallery.
- **size** (*int*) – Size of the gallery.

#### Request Headers

- **Host** – `portal.id.nec.com.au`
- **Accept** – `application/json`
- **Content-Type** – `application/json`
- **x-amz-date** – UTC timestamp using ISO 8601 format: `YYYYMMDD'T'HHMMSS'Z'`.
- **Authorization** – AWS Signature Version 4.

#### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "galleryguid-28da-4dd4-b10b-e48c6be09689",
  "name": "Y10Students",
  "description": "Year 10 Students",
  "size": 2000
}
```

### Response JSON Object

- **id** (*string*) – Gallery id.
- **name** (*string*) – Name.
- **description** (*string*) – Description.
- **size** (*int*) – Size.

### Response Headers

- **Content-Type** – application/json

### Status Codes

- **200 OK** – Gallery updated.

---

**Note:** Gallery size can only be increased.

---

## Delete Gallery

**DELETE** /api/galleries/ (**string:** *galleryId*)

Deletes a gallery and its related applications.

### Example request:

```
DELETE /api/galleries/galleryguid-28da-4dd4-b10b-e48c6be09689 HTTP/1.1
Host: portal.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

### Parameters

- **galleryId** – Gallery id.

### Request Headers

- **Host** – portal.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

### Example response:

```
HTTP/1.1 204 No Content
Content-Type: application/json
```

### Response Headers

- **Content-Type** – application/json

### Status Codes

- 204 No Content – Gallery deleted.

## 1.3.2 Applications

Manage tenant applications.

### List Applications

#### GET /api/applications

Retrieve all applications.

#### Example request:

```
GET /api/applications HTTP/1.1
Host: portal.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

#### Request Headers

- **Host** – portal.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

#### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "applications": [
    {
      "id": "applicationId01",
      "name": "HR System",
      "apiKey": [Application API key]
    },
    {
      "id": "applicationId02",
      "name": "Building Access System",
      "apiKey": [Application API key]
    }
  ]
}
```

#### Response JSON Object

- **applications** (*array*) – Containing **id** (*string*): Application id, **name** (*string*): Name, **apiKey** (*string*): Application API Key.

#### Response Headers

- **Content-Type** – application/json

**Status Codes**

- **200 OK** – OK.

**Get Application**

**GET** /api/applications/ (**string:** *applicationId*)

Retrieve application details.

**Example request:**

```
GET /api/applications/applicationId01 HTTP/1.1
Host: portal.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

**Parameters**

- **applicationId** – Application id.

**Request Headers**

- **Host** – portal.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "applicationId01",
  "name": "HR System",
  "description": "HR System",
  "apiKey": [Application API key],
  "faceMinimumQualityScore": 0.6
}
```

**Response JSON Object**

- **id** (*string*) – Gallery id.
- **name** (*string*) – Name.
- **description** (*string*) – Description.
- **apiKey** (*string*) – Application API Key.

**Request JSON Object**

- **faceMinimumQualityScore** (*float*) – Minimum estimated overall quality of face - see *Face Attributes*.

#### Response Headers

- **Content-Type** – application/json

#### Status Codes

- 200 OK – OK.

## Create Application

### POST /api/applications

Create an application.

#### Example request:

```
POST /api/applications HTTP/1.1
Host: portal.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "name": "Building Access System",
  "description": "Building Access System",
  "faceMinimumQualityScore": 0.8,
  "galleryId": "galleryguid-327D-4D31-8E25-35891A034220",
}
```

#### Request JSON Object

- **name** (*string*) – Name of the application.
- **description** (*string*) – Description of the application.
- **faceMinimumQualityScore** (*float*) – Minimum estimated overall quality of face - see *Face Attributes*, must be greater than or equal to 0.45.
- **galleryId** (*string*) – Id of the target gallery.

#### Request Headers

- **Host** – portal.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

#### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
```

(continues on next page)



(continued from previous page)

```

{id": "applicationId02",
"name": "Building Access System",
"description": "Building Access System",
"apiKey": [Application API key],
"faceMinimumQualityScore": 0.8
}

```

**Response JSON Object**

- **id** (*string*) – Application id.
- **name** (*string*) – Name.
- **description** (*string*) – Description.
- **apiKey** (*string*) – Application API Key.

**Request JSON Object**

- **faceMinimumQualityScore** (*float*) – Minimum estimated overall quality of face - see *Face Attributes*.

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- **200 OK** – Application created.

**Update Application****PATCH** /api/applications/ (**string**: *applicationId*)

Update an application.

**Example request:**

```

PATCH /api/applications HTTP/1.1
Host: portal.id.nec.com.au
Accept: application/json
Content-Type: application/json
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]

{
  "name": "Building Access System",
  "description": "Building Access System",
  "faceMinimumQualityScore": 0.6
}

```

**Parameters**

- **applicationId** – Application id.

**Request JSON Object**

- **name** (*string*) – Name of the application.
- **description** (*string*) – Description of the application.

- **faceMinimumQualityScore** (*float*) – Minimum estimated overall quality of face - see *Face Attributes*, must be greater than or equal to 0.45.

#### Request Headers

- **Host** – portal.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

#### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "applicationId02",
  "name": "Building Access System",
  "description": "Building Access System"
  "apiKey": [Application API key],
  "faceMinimumQualityScore": 0.8
}
```

#### Response JSON Object

- **id** (*string*) – Application id.
- **name** (*string*) – Name.
- **description** (*string*) – Description.
- **apiKey** (*string*) – Application API Key.

#### Request JSON Object

- **faceMinimumQualityScore** (*float*) – Minimum estimated overall quality of face - see *Face Attributes*.

#### Response Headers

- **Content-Type** – application/json

#### Status Codes

- **200 OK** – Application updated.

## Delete Application

**DELETE** /api/applications/ (**string:** *applicationId*)

Deletes an application.

#### Example request:

```
DELETE /api/applications/applicationid02 HTTP/1.1
Host: portal.id.nec.com.au
Accept: application/json
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
x-amz-date: [YYYYMMDD'T'HHMMSS'Z' UTC timestamp]
Authorization: [AWS Signature Version 4]
```

**Parameters**

- **applicationId** – Application id.

**Request Headers**

- **Host** – portal.id.nec.com.au
- **Accept** – application/json
- **Content-Type** – application/json
- **x-amz-date** – UTC timestamp using ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'.
- **Authorization** – AWS Signature Version 4.

**Example response:**

```
HTTP/1.1 204 No Content
Content-Type: application/json
```

**Response Headers**

- **Content-Type** – application/json

**Status Codes**

- **204 No Content** – Application deleted.

## 1.4 Error Handling

In addition to the valid successful and unsuccessful status codes listed for each endpoint, exceptions will result in a error response (400 or 500 status code) with a JSON Object containing the error details.

For example, a request to **register** endpoint which does not include the mandatory **face** property, will return the following:

**POST /v1.1/subjects**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "message": "face property must be provided.",
  "errorCode": 1001,
  "errorType": "InvalidRequest"
}
```

**Response JSON Object**

- **message** (*string*) – Description of the error.
- **errorCode** (*int*) – Numeric code representing the error. See *Error Codes and Types*.

- **errorType** (*string*) – Named constant representing the error. See *Error Codes and Types*.

**Response Headers**

- Content-Type – application/json

**Status Codes**

- 400 Bad Request – Client exception.
- 500 Internal Server Error – Server exception.

### 1.4.1 Error Codes and Types

Type	Code	Description
InvalidRequest	1001	Request is invalid, possible missing parameters.
Duplicate	1002	Duplicate request, identifier in message.
Timeout	1003	Request timed out.
NoFace	2001	No face was found in the provided image.
PoorQuality	2002	The overall <code>faceQualityScore</code> was too low to process.
MultipleFaces	2003	Multiple faces were found in the image. Only one is permitted for this type of request.

### 2.1 Platforms

NEC ID provides SDKs for a number of platforms:

- Android
- iOS (Objective-C)
- iOS (Swift)
- Java
- JavaScript
- Ruby



---

## HTTP Routing Table

---

### /api

9  
DELETE /v1.1/subjects/(string:subjectId)/events/(string:eventId), 14  
DELETE /v1.1/tags/(string:name), 23  
GET /api/applications, 34  
GET /api/applications/(string:applicationId), 35  
GET /api/galleries, 28  
GET /api/galleries/(string:galleryId), 29  
POST /api/applications, 36  
POST /api/galleries, 30  
DELETE /api/applications/(string:applicationId), 38  
DELETE /api/galleries/(string:galleryId), 33  
PATCH /api/applications/(string:applicationId), 37  
PATCH /api/galleries/(string:galleryId), 32

### /v1.1

GET /v1.1/jobs/bulkregister/(string:batchId)/(string:pagingId), 25  
GET /v1.1/subjects, 4  
GET /v1.1/subjects/(string:subjectId)/events, 10  
GET /v1.1/tags, 21  
POST /v1.1/face/extract, 15  
POST /v1.1/face/search, 17  
POST /v1.1/face/verify, 20  
POST /v1.1/jobs/bulkregister, 24  
POST /v1.1/subjects, 39  
POST /v1.1/subjects/(string:subjectId)/events, 11  
POST /v1.1/tags/(string:name), 22  
PUT /v1.1/subjects/(string:subjectId), 8  
PUT /v1.1/subjects/(string:subjectId)/events/(string:eventId), 13  
PUT /v1.1/tags/(string:oldName), 22  
DELETE /v1.1/subjects/(string:subjectId),